Universidade Federal Fluminense
Escola de Engenharia
Curso de Graduação em Engenharia de Telecomunicações

Bernardo Prado de Abreu

Aplicações de processamento de sinais em extração de informação musical

Niterói – RJ

2021

Bernardo Prado de Abreu

Aplicações de processamento de sinais em extração de informação musical

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Engenheiro de Telecomunicações.

Orientador: Prof. Dr. Tadeu Nagashima Ferreira
Coorientador: Prof. Dr. Luiz Wagner Pereira Biscainho

Niterói – RJ
2021

.

A162a Abreu, Bernardo Prado de
Aplicações de processamento de sinais em extração de informação musical / Bernardo Prado de Abreu ; Tadeu Nagashima Ferreira, orientador ; Luiz Wagner Pereira Biscainho, coorientador. Niterói, 2021.
54 f. : il.

Trabalho de Conclusão de Curso (Graduação em Engenharia de Telecomunicações)-Universidade Federal Fluminense, Escola de Engenharia, Niterói, 2021.

1. Processamento de sinais. 2. Som (Engenharia). 3. Produção intelectual. I. Ferreira, Tadeu Nagashima, orientador. II. Biscainho, Luiz Wagner Pereira, coorientador. III. Universidade Federal Fluminense. Escola de Engenharia. IV. Título.

CDD -

.

Bernardo Prado de Abreu

Aplicações de processamento de sinais em extração de informação musical

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Telecomunicações da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Engenheiro de Telecomunicações.

Aprovada em 30 de abril de 2021.

BANCA EXAMINADORA

Prof. Dr. Tadeu Nagashima Ferreira - Orientador
Universidade Federal Fluminese - UFF

Prof. Dr. Luiz Wagner Pereira Biscainho - Coorientador
Universidade Federal do Rio de Janeiro - UFRJ

Prof. Dr. Alexandre Santos de la Vega
Universidade Federal Fluminense - UFF

Prof. Dr. Edson Luiz Cataldo Ferreira
Universidade Federal Fluminense - UFF

Niterói – RJ

2021

Resumo

Extração de Informação Musical (MIR, do inglês *Music Information Retrieval*) é uma área de pesquisa interdisciplinar relacionada com diversos assuntos como, por exemplo, musicologia, psicoacústica, processamento de sinais e aprendizado de máquina. Essa área trata de um conjunto de técnicas que tem como objetivo a obtenção de dados musicalmente relevantes a partir de sinais de áudio. Neste trabalho, foram explorados alguns tópicos importantes para esse assunto. Primeiramente, foi abordada uma maneira de representar um sinal de música, facilitando assim a sua posterior análise. Depois, foram estudados algoritmos de detecção de *onset* e rastreamento de frequência fundamental, que buscam dados de grande utilidade para outras aplicações de processamento de sinais de música. Para melhorar o desempenho desses algoritmos, foi implementada uma etapa de pré-processamento no sinal, utilizando um filtro de mediana móvel. O impacto desse pré-processamento foi avaliado por meio de experimentos, que indicaram uma influência positiva do filtro nos algoritmos.

Palavras-chave: Extração de informação musical. Representação tempo-frequencial. Detecção de *onset*. Rastreamento de frequência fundamental. Filtragem por mediana.

Abstract

Music Information Retrieval (MIR) is an interdisciplinary research area related to many different disciplines, such as musicology, psychoacoustics, signal processing and machine learning. This field deals with a set of techniques that aims to obtain musically relevant data from audio signals. In this work, a few important topics in this subject were explored. First, we addressed a way of representing a music signal in order to facilitate analysis. Then, we studied some algorithms regarding onset detection and fundamental frequency tracking, which seek data of great utility for other applications on music signal processing. In order to improve the performance of those algorithms, we implemented a preprocessing step, using a moving median filter. The impact of this preprocessing step was evaluated through a set of experiments, which indicated a positive influence from the filter on the algorithms.

Keywords: Music information retrieval. Time-frequency representation. Onset detection. Fundamental frequency tracking. Median filtering.

Agradecimentos

À minha família, por todo o apoio que me deram ao longo da minha vida.

Aos meus orientadores Tadeu Nagashima Ferreira e Luiz Wagner Pereira Biscainho, por tudo que me ensinaram e também pela paciência que tiveram comigo durante todo o tempo desse trabalho.

À minha namorada Thaysa, por estar ao meu lado nos momentos mais difíceis.

Aos meus amigos Matheus e Bruna, que me acompanharam durante toda a trajetória da faculdade.

Aos professores do curso de Engenharia de Telecomunicações da Universidade Federal Fluminense, pelos ensinamentos profissionais e pessoais. Em especial, aos professores Alexandre Santos de la Vega e Jacqueline Silva Pereira.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

1.1 Motivações

Música é uma das principais formas de manifestação da cultura humana, além de ser um aspecto essencial da identidade de muitos grupos na nossa sociedade. Por meio dessa arte, é possível expressar os mais diversos sentimentos, tendo assim grande significado na vida de muitas pessoas.

Além de sua importância cultural, a música tem grande valor econômico. Com o desenvolvimento de técnicas de gravação, compressão e sistemas de armazenamento digital de músicas, podemos observar um aumento significativo na produção de peças musicais. Como resultado, temos um grande volume de dados armazenados, que correspondem não apenas aos sinais, mas também a informações a respeito das peças e dos artistas. Esses dados são especialmente úteis para serviços de *streaming* de música, que utilizam sistemas de recomendação baseados no tipo de conteúdo consumido pelos usuários.

Para que esses sistemas de recomendação funcionem corretamente, é necessário que os sinais sejam devidamente catalogados de acordo com seus atributos, como, por exemplo, o gênero musical. A obtenção manual desses atributos pode ser particularmente difícil, além de ser um trabalho entediante e sujeito a erros. Por essa razão, o desenvolvimento de técnicas que permitam a extração automática de dados a partir de um sinal de música é um assunto que tem ganhado grande visibilidade nas últimas décadas. Nesse contexto, existe uma área de pesquisa que tem se destacado, chamada de Extração de Informação Musical (MIR, do inglês *Music Information Retrieval*), que inclui uma variedade de conhecimentos em diferentes áreas, como musicologia, psicoacústica, teoria da informação, processamento

de sinais, aprendizado de máquina, entre outras.

Em especial, o ritmo e a melodia são aspectos muito importantes de uma música, havendo assim uma grande demanda por dados relacionados a essas características. Por essa razão, diversas técnicas foram propostas para que tais atributos sejam extraídos de forma automática e com maior eficiência. Sendo assim, é de grande importância o estudo dos algoritmos que desempenham essas tarefas. Neste trabalho, serão estudadas algumas características gerais da música, com um destaque para a música ocidental.

1.2 Objetivo

O principal objetivo desse trabalho é reunir um conjunto de conceitos básicos e técnicas, dentro do contexto de "Extração de Informação Musical". Serão abordados alguns dos desafios nessa área, além de soluções simples, porém efetivas, para esses problemas. Dessa forma, espera-se que seja formada uma base de conhecimento, a partir da qual outros estudos possam ser realizados. Também faz parte desse trabalho elaborar uma proposta para melhorar o desempenho de algumas das técnicas abordadas.

1.3 Organização do trabalho

Este trabalho segue com o Capítulo 2, que trata de uma maneira de representar um sinal que é bastante útil para realizarmos análises e extrair seus atributos. Nesse capítulo, serão apresentados três métodos para obtermos essa representação.

Em seguida, no Capítulo 3, é abordado o tópico de detecção de *onsets*. São descritas a definição dessa tarefa, uma implementação bastante utilizada, além de melhorias propostas para esse algoritmo.

O Capítulo 4 fala sobre rastreamento de frequência fundamental, que está relacionado com a extração da melodia principal de uma peça musical. São feitas algumas definições e, então, é apresentado um algoritmo para a realização dessa tarefa.

Depois, no Capítulo 5, é proposto um pré-processamento no sinal a fim de melhorar o desempenho dos algoritmos descritos nos Capítulos 3 e 4.

Para avaliar a eficiência do método proposto, são realizados experimentos. Estes são descritos no Capítulo 6, bem como os resultados obtidos.

Por fim, o Capítulo 7 apresenta conclusões e sugere possíveis trabalhos futuros.

Capítulo 2

Representação tempo-frequencial

Sinais de áudio são comumente representados por sua forma de onda, o que permite a observação de seu comportamento ao longo do tempo. Esta é chamada de representação no domínio temporal e pode ser utilizada para obter informações a respeito de eventos bem localizados no tempo, como o instante em que uma nota musical é tocada, por exemplo. Porém, existem outros tipos de informação que não podem ser obtidos de maneira prática a partir da análise do sinal no tempo. Nesse caso, é interessante utilizar uma representação que mostre a composição em frequências do mesmo. Isso torna possível identificar, entre outras coisas, quais instrumentos estão presentes em uma dada composição musical. Uma forma muito conhecida de se obter tal composição é a Transformada de Fourier [3], que gera a representação no domínio da frequência.

Contudo, a Transformada de Fourier não leva em consideração o fato de que sinais de áudio não são estacionários. A ocorrência de eventos musicais, por exemplo, está relacionada a mudanças na composição espectral ao longo do tempo. Portanto, a representação no domínio da frequência do sinal inteiro não é suficiente para analisar um sinal de áudio de maneira adequada.

É necessário, portanto, utilizar uma outra representação que permita identificar as componentes espectrais do sinal em um determinado segmento do mesmo. Para isso, devemos dividir o sinal em diversos blocos, ou quadros de tempo, o que nos permitirá assumir que, dentro de um quadro, o sinal é, em boa aproximação, estacionário. Isso pode ser feito a partir da utilização de funções conhecidas como janelas. Tais funções são caracterizadas por possuírem valores não-nulos somente em um dado intervalo [1]. Ao realizar a Transformada de Fourier de um sinal janelado, obtêm-se as frequências presentes

em cada intervalo, como desejado. Para observar a evolução das componentes espectrais ao longo do tempo, basta repetir esse procedimento em diferentes instantes do sinal.

Dessa forma, é possível gerar uma representação tempo-frequencial, que será o ponto de partida das técnicas estudadas neste trabalho. Neste capítulo, serão apresentadas as representações mais utilizadas na análise de sinais de música.

2.1 Transformada de Fourier de tempo curto

A Transformada de Fourier de um sinal analógico $x(t)$ é definida como

$$X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt.$$

Como os sinais a serem analisados são digitais, é necessária uma maneira mais adequada de realizar esse procedimento. Para isso, assumindo que o sinal tem duração limitada, o método mais comum é o da Transformada Discreta de Fourier (DFT, do inglês *Discrete Fourier Transform*) [2], que é definida da seguinte maneira:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn},$$

onde k é o índice de frequência e N é o número de amostras utilizadas no cálculo da DFT.

Para obtermos a representação tempo-frequencial, multiplicamos o sinal por uma janela $w[n]$ e realizamos diversas DFTs, deslocando a janela no tempo a cada iteração. Assim, define-se a Transformada de Fourier de Tempo Curto (STFT, do inglês *Short Time Fourier Transform*). A partir deste cálculo, é possível gerar o espectrograma de potência, tomando o módulo da STFT ao quadrado. A representação fica então definida da seguinte maneira:

$$X[m,k] = \left| \sum_{n=0}^{N-1} x[n] w[n - hm] e^{-j\frac{2\pi}{N}kn} \right|^2,$$

onde m *2* N é o índice temporal da STFT e h *2* N é o tamanho, em amostras, do salto da janela.

A multiplicação do sinal pela janela resulta em uma nova sequência. No domínio da frequência, isso significa que o resultado obtido pela DFT será a convolução circular do espectro do sinal com o da janela [3]. Portanto, é importante observar a influência que a janela exerce na representação e, assim, escolher adequadamente os parâmetros, de forma

a torná-la mais precisa, tanto no tempo quanto na frequência. Em ambos os casos, utiliza-se a esparsidade como medida de precisão. Uma representação é considerada esparsa no tempo quando consegue diferenciar eventos localizados no tempo, como, por exemplo, toques em um instrumento percussivo. No domínio da frequência, a representação é considerada esparsa quando possui, em suas componentes espectrais, picos proeminentes e, no entorno de cada pico, valores aproximadamente nulos.

Em um primeiro momento, pode-se pensar que um bom formato de janela é o retangular, pois não provoca alterações no sinal durante o intervalo de interesse e é de fácil implementação. Porém, as descontinuidades existentes nas bordas provocam o surgimento de componentes ao longo do espectro, como pode ser visto na Figura 2.1. Essas componentes causam alterações na representação, dificultando consideravelmente a identificação das componentes de menor intensidade do sinal. Uma forma de amenizar esse problema é utilizar janelas com uma transição mais suave nas bordas. Em processamento de áudio, são comumente utilizadas as janelas de Hann e Hamming [1].

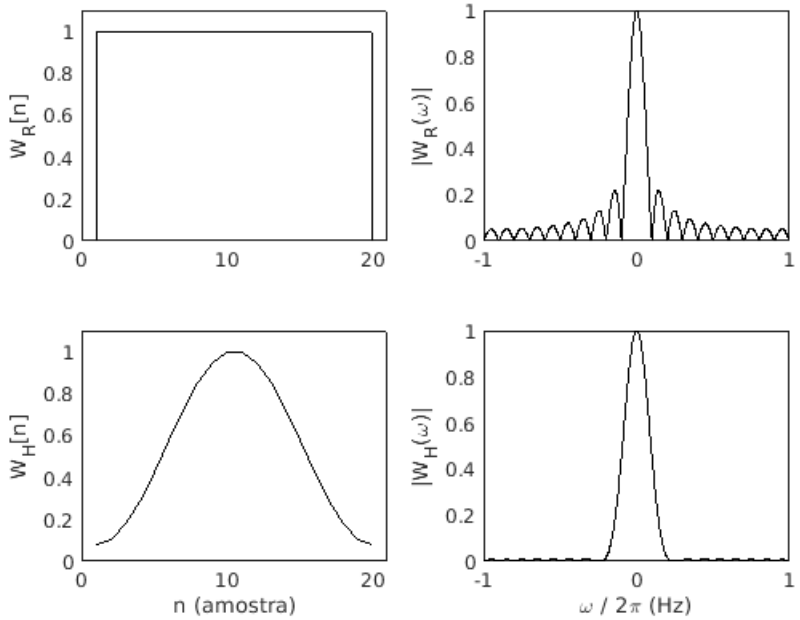

Figura 2.1: Janela retangular e janela de Hamming com comprimento de 20 amostras com suas respectivas transformadas de Fourier.

Outro parâmetro importante a ser verificado é o tamanho da janela, que está diretamente relacionado com as resoluções no tempo e na frequência. Da propriedade de escalamento da Transformada de Fourier, temos que uma redução da janela no tempo

implica um alargamento em seu espectro. Isso significa que as componentes do sinal, ao serem convoluídas, ficarão espalhadas em torno de seus valores originais. Isso é entendido como uma menor resolução em frequência. Por outro lado, o uso de janelas curtas no tempo resulta em um menor trecho do sinal sendo analisado. Dessa forma, as variações na composição espectral serão menores dentro deste quadro, o que leva a uma maior resolução temporal. Em um raciocínio análogo, conclui-se que janelas de longa duração produzem uma maior resolução em frequência e menor resolução no tempo. Nota-se, então uma relação inversa entre as resoluções em função do tamanho da janela. Tal relação é formalmente definida pelo Princípio da Incerteza, que mostra que é impossível obter, simultaneamente, uma precisão arbitrariamente alta no tempo e na frequência [1]. As Figuras 2.2 e 2.3 mostram o resultado do espectrograma de um sinal gerado por um trompete solo amostrado a uma taxa de 44,1 kHz, utilizando janelas de diferentes tamanhos.

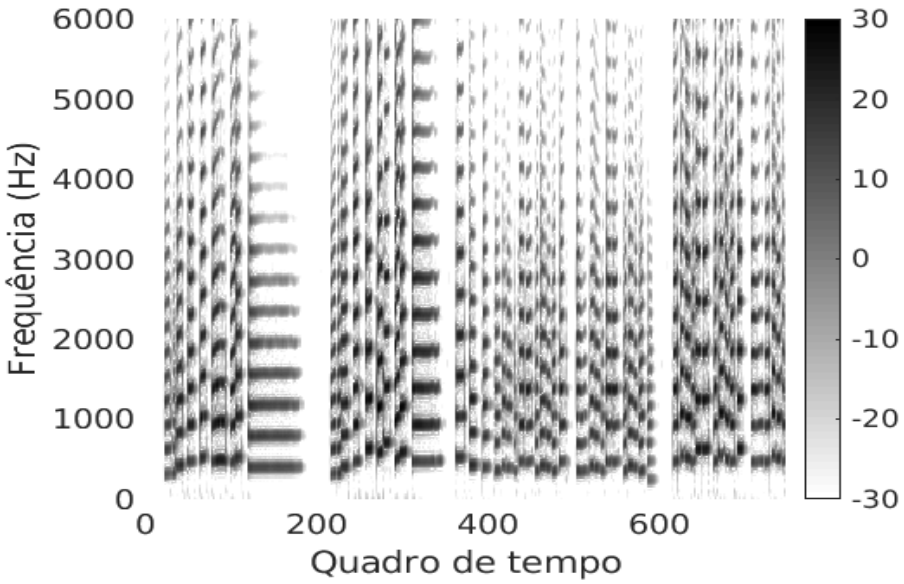

Figura 2.2: Espectrograma do sinal de exemplo, para janela de 20 ms (882 amostras).

Ao considerarmos o Princípio da Incerteza, é possível concluir que o tamanho da janela pode impactar o resultado da análise. Um espectrograma gerado por uma janela pequena torna mais fácil a observação do comportamento do sinal ao longo do tempo. Porém, é mais difícil reconhecer com precisão as frequências que, de fato, compõem o sinal, já que a resolução frequencial é menor. Janelas maiores, por outro lado, produzem o efeito contrário, sendo melhores na identificação das frequências, enquanto tornam a

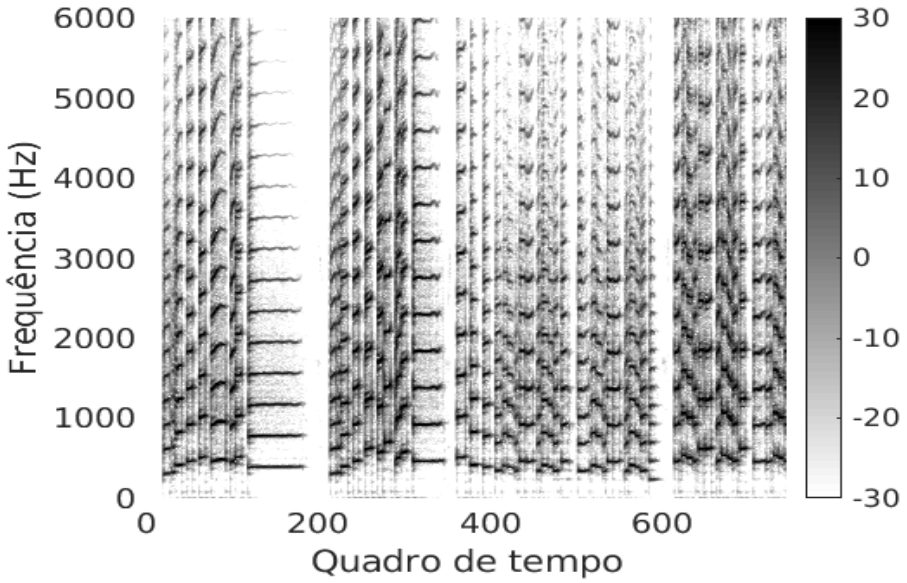

Figura 2.3: Espectrograma do sinal de exemplo, para janela de 100 ms (4410 amostras).

análise temporal menos acurada.

Fica claro, portanto, que há limitações no uso dessa representação. Além disso, serão apresentados, nas próximas seções deste capítulo, outros problemas associados à STFT na análise de sinais de música, bem como maneiras de contorná-los, com o objetivo de tornar a representação melhor.

2.2 Transformada de Q constante

A escala mais utilizada na música ocidental, chamada de escala de temperamento igual, utiliza notas musicais com frequências fundamentais espaçadas segundo uma progressão geométrica. Por conta disso, as notas presentes em uma peça musical podem não ser acuradamente representadas pela DFT, já que esta mapeia as frequências de forma linear [4]. Para resolver esse problema, foi criada uma variação da STFT: a Transformada de Q Constante (CQT ou *Constant Q Transform*), que possui uma resolução frequencial proporcional à frequência central de cada raia frequencial (que pode ser associada a um filtro nela centrado), permitindo assim um mapeamento mais homogêneo das notas musicais no espectro [4]. O termo "Q constante" está atribuído ao fator de qualidade Q do filtro associado, definido como a razão entre cada frequência e a resolução frequencial:

$$Q = \frac{f_k}{\Delta f}. \tag{2.1}$$

As frequências da CQT são distribuídas da seguinte maneira:

$$f_k = f_{min} 2^{\frac{k}{B}},$$

onde $f_{min} = f_0$ é uma frequência mínima estabelecida, B é o número de componentes por oitava e f_k é a k-ésima componente, onde $f_{min} < f_k < f_{max}$ e a frequência f_{max} é menor do que a frequência de Nyquist.

Assim, a resolução frequencial, definida como a distância entre componentes adjacentes, é calculada como

$$\begin{aligned} \Delta f_{CQT} &= f_{k+1} \quad f_k \\ &= f_{min} 2^{\frac{k+1}{B}} \quad f_{min} 2^{\frac{k}{B}} \\ &= f_{min} 2^{\frac{k}{B}} (2^{\frac{1}{B}} \quad 1) \\ &= f_k (2^{\frac{1}{B}} \quad 1). \end{aligned}$$

Substituindo f_k e Δf na Equação (2.1), temos

$$Q = \frac{f_k}{f_k (2^{\frac{1}{B}} \quad 1)} = (2^{\frac{1}{B}} \quad 1) \ ^1,$$

mostrando assim que o fator Q é constante.

Foi visto na Seção 2.1 que a resolução frequencial da STFT possui uma relação inversa com o tamanho da janela. De forma mais específica, pode-se demonstrar [4] que o seu valor é:

$$\Delta f_{STFT} = \frac{F_s}{N},$$

onde F_s é a frequência de amostragem do sinal e N é o tamanho da janela, em amostras. Para que o fator Q se mantenha constante, é necessário que cada componente espectral seja associada a um tamanho de janela. Portanto, na CQT, o parâmetro N passa a ser um conjunto N_k tamanhos de janela, que são calculados ao substituirmos a resolução frequencial da STFT na Equação (2.1):

$$N_k = Q \frac{Fs}{f_k}.$$

Fazendo, então, as devidas substituições, chegamos à definição da CQT, dada por:

$$X_{CQT}[k] = \frac{1}{N_k} \sum_{n=0}^{N_k} x[n] w_k[n] e^{-j2\pi n \frac{Q}{N[k]}},$$

onde $w_k[n]$ é uma janela de tamanho N_k [4].

É possível notar, no entanto, que essa representação possui um alto custo computacional, já que é necessário repetir o cálculo da DFT para cada tamanho de janela em cada segmento do sinal. Além disso, o uso de janelas diferentes resulta em uma resolução temporal diferente em cada faixa de frequências, já que, para cada componente, será analisado um trecho do sinal de tamanho específico. Essas duas características da CQT devem ser levadas em consideração na escolha dos parâmetros de resolução e quantidade de frequências utilizadas na análise.

2.3 Transformada *Fan Chirp*

A STFT parte da ideia de que, dentro da janela, um sinal pode ser considerado estacionário. Porém, no caso de sinais de música, esta não é sempre uma boa aproximação. Não é incomum a existência de trechos onde a variação na composição espectral é considerável, mesmo dentro da janela. Além disso, por conta da natureza harmônica desses sinais, as variações são ainda maiores nas componentes de mais alta frequência. Isso resulta em uma baixa resolução frequencial.

Uma forma de representar sinais harmônicos não-estacionários é a Transformada *Fan Chirp* (FChT, do inglês *Fan Chirp Transform*), inicialmente proposta em [5]. Essa transformada consiste na decomposição do sinal em uma base de funções cuja frequência varia no tempo. Assim, a aproximação utilizada poderá se adequar ao tipo de sinal. Para o caso de sinais de música, é sugerida uma formulação que considera variações lineares, tomando como referência a frequência fundamental do sinal em um dado quadro de tempo. Dessa forma, as funções de base escolhidas serão senoides com variação linear no tempo, conhecidas como *chirps* lineares [6]. Então, a Transformada Fan Chirp é definida da seguinte forma:

$$X(f, \alpha) = \int_{-\infty}^{\infty} x(t) \phi_\alpha^0(t) e^{-j2\pi f \phi_\alpha(t)} dt, \tag{2.2}$$

onde $\phi_\alpha(t)$ é uma função dada por

$$\phi_\alpha(t) = \left(1 + \frac{1}{2}\alpha t\right) t$$

e α é o coeficiente angular do *chirp*.

Da função ϕ_α, nota-se que essa transformada é uma forma modificada da Transformada de Fourier, que pode ser obtida quando $\alpha = 0$.

Fazendo uma mudança de variável $\tau = \phi_\alpha(t)$, a Equação (2.2) torna-se:

$$X(f, \alpha) = \int_{1/\alpha}^{1} x(\phi_\alpha{}^{1}(\tau)) e^{\ j2\pi f \tau} d\tau,$$

onde $\phi_\alpha{}^{1}(t)$ é dada por

$$\phi_\alpha{}^{1}(t) = \frac{1}{\alpha} + \frac{\sqrt{1 + 2\alpha t}}{\alpha}.$$

Essa formulação é a Transformada de Fourier de um sinal deformado no tempo pela função $\phi_\alpha{}^{1}$. Além disso, é necessário assumir que $x(t) = 0$ para $t \quad 1/\alpha$, a fim de evitar *aliasing* [6]. Isso significa que, em sua versão discreta, o cálculo da FChT poderá ser feito por meio de algoritmos mais rápidos, como a FFT.

Para gerar a representação tempo-frequencial, é necessário realizar o janelamento do sinal. Porém, como o sinal sofre deformações no tempo, uma deformação na janela $w(t)$ também deverá ser aplicada. Dessa forma, fica definida a Transformada Fan Chirp de Tempo Curto (STFChT, ou *Short Time Fan Chirp Transform*):

$$X_w(f, \alpha) = \int_{1}^{1} x(t) w(\phi_\alpha(t)) \phi'_\alpha(t) e^{\ j2\pi f \phi_\alpha(t)} dt. \tag{2.3}$$

Agora, o sinal é modelado como um conjunto de *chirps* lineares para cada segmento. Isso significa que, em cada quadro de tempo, deverá ser encontrado um valor ótimo de α. Para encontrar o valor desse coeficiente, geralmente é feita uma busca exaustiva dentro de um conjunto pré-determinado. É escolhido aquele α que gera a representação com maior esparsidade.

Encontrados os valores de α, basta calcular a função de deformação temporal ϕ_α e aplicá-la na Equação (2.3). Esse procedimento, em sinais discretos, é feito por meio de uma interpolação [6].

É importante notar que essa transformada possui um alto custo computacional, especialmente por conta da busca pelo valor de α. Além disso, esse método é capaz de

representar apenas uma fonte com boa esparsidade, não sendo muito eficiente na representação de sinais polifônicos.

Capítulo 3

Detecção de *onset*

Um componente muito importante da música é a sua estrutura temporal, em especial o ritmo e o andamento. O ritmo traz uma sensação de regularidade e está diretamente relacionado com o gênero musical, enquanto o andamento está associado à velocidade com que uma peça é tocada. Esses aspectos são fundamentais e a extração automática de tais informações é um assunto central na área de processamento de sinais de música [3].

Existem diversas maneiras de realizar essa tarefa e, na maioria delas, uma etapa essencial é a extração dos instantes de ocorrência de eventos musicais [1]. Esta tarefa é chamada de detecção de *onset*. Neste capítulo, serão apresentadas as principais etapas do processo de detecção de *onset*: redução, seleção de picos e avaliação dos resultados. Também é possível incluir uma etapa de pré-processamento, que será explicada no Capítulo 5.

3.1 Redução

O objetivo da etapa de redução é transformar o sinal em uma função mais simples, conhecida como função de detecção. Em geral, essa função representa mudanças em atributos do sinal e possui máximos locais que coincidem com os instantes de ocorrência dos *onsets* [9].

Em [8], são estudados diversos métodos de redução, que podem ser classificados de acordo com qual atributo do sinal será considerado. É importante notar que não existe um algoritmo ótimo para todos os tipos de música. Isso porque os resultados dependem não somente da adequação de parâmetros, mas também do tipo de sinal a ser analisado.

As notas produzidas por um piano, por exemplo, possuem um transitório bem definido, enquanto um violino pode gerar sequências de notas de forma mais suave.

Dentre os métodos existentes para localizar esses transitórios, o fluxo espectral é um dos mais utilizados. Inicialmente apresentado em [7], este algoritmo se destaca pela fácil implementação e baixa complexidade computacional. Seu cálculo consiste, inicialmente, na obtenção da representação tempo-frequencial do sinal, realizada por meio da STFT. A partir do módulo do espectrograma, é calculada a diferença entre as componentes espectrais ao longo do tempo. Em seguida, somam-se os resultados de cada raia espectral. Isso mostra a variação na energia das componentes espectrais. A formulação mais popular é feita em [9], como segue:

$$SF[n] = \sum_{k=\ \frac{N}{2}}^{\frac{N}{2}\ \ 1} H[jX[n,k]j \quad jX[n \quad 1,k]j],$$

onde $H[x] = \frac{x+jxj}{2}$ é uma função de retificação de meia onda. Os valores negativos são desconsiderados porque, na ocorrência de um *onset*, é esperado um aumento na energia, que aparece como um valor positivo no fluxo espectral. A Figura 3.1 mostra o resultado da etapa de redução, aplicada ao espectrograma visto na Figura 2.3.

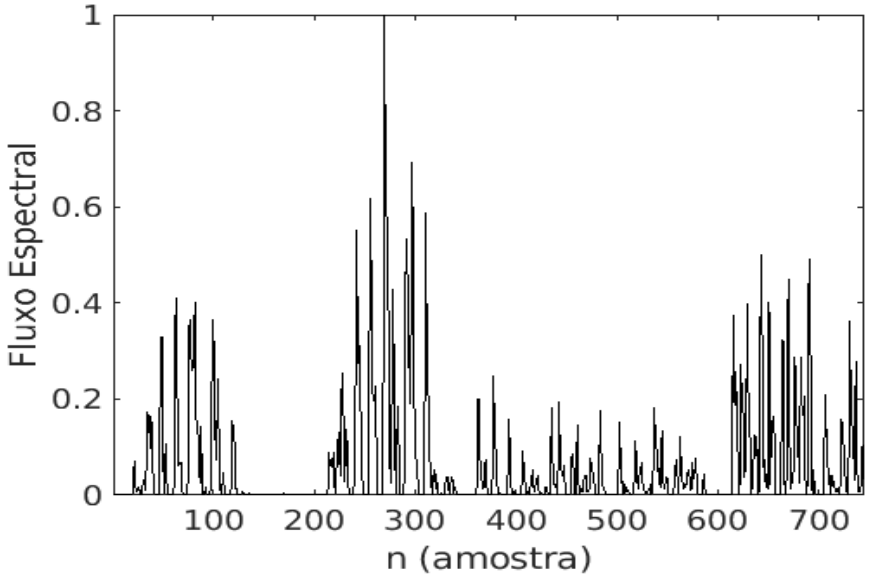

Figura 3.1: Fluxo Espectral do sinal de exemplo.

Além dos passos citados, é possível adicionar etapas ao algoritmo, como uma decimação no sinal ou uma compressão logarítmica no espectrograma. Também devem ser levados em consideração parâmetros da STFT, como o tamanho da janela e a distância entre janelas consecutivas. Em [12], é feita uma avaliação do desempenho do algoritmo, levando em consideração diversas modificações e variações de parâmetros.

3.2 Seleção de picos

Como foi dito na Seção 3.1, *onsets* aparecem na função de detecção como máximos locais (picos). Porém, nem todos os picos da função são relevantes. Alguns podem ser gerados não apenas por ruídos do sinal, mas também por outros eventos musicais, como um vibrato, por exemplo [8]. Portanto, selecionar todos os picos não é uma boa estratégia, pois irá resultar em um grande número de falsos positivos. Para tornar o processo de detecção mais acurado, é necessário que o algoritmo de seleção de picos seja mais robusto. Isso pode ser feito ao escolhermos apenas os picos de maior relevância.

Para isso, é importante que a função seja, primeiramente, normalizada. Ao comprimir os valores para um determinado intervalo (geralmente de 0 a 1), é possível utilizar parâmetros semelhantes para as próximas etapas, independente da faixa dinâmica da função de detecção. Em [8] e [9], é sugerido que a função seja normalizada de tal forma que seu valor médio seja zero e que o desvio padrão seja unitário.

Depois, é aplicado um limiar, que é um valor mínimo para que um pico seja levado em consideração. A forma mais simples de se realizar esta tarefa é utilizando um valor constante. Essa abordagem, porém, não é das melhores. Isso porque os sinais de música geralmente apresentam variações consideráveis em sua intensidade. Assim sendo, um limiar fixo tenderia a ignorar *onsets* em algumas passagens e detectar picos irrelevantes em outras.

É interessante, portanto, utilizar um limiar capaz de se adaptar às mudanças na dinâmica do sinal. De maneira geral, esse limiar é uma versão suavizada da função de detecção. Uma forma simples de suavizar a função é aplicar um filtro de média móvel. Porém, esse método produz distorções na região em torno dos picos de maior intensidade, o que prejudica a detecção dos picos nessa região [8].

Para contornar esse problema, é desejável que sejam utilizados métodos capazes de

"ignorar" picos. Em [11], é sugerida uma técnica chamada de SSE (*Stochastic Spectrum Estimation*). O algoritmo é o seguinte, para uma função qualquer $S[n]$:

1. Calcular $R[n] = \frac{1}{S[n]}$;
2. Aplicar o filtro de média móvel em $R[n]$, obtendo $\widetilde{R}[n]$;
3. Inverter o resultado obtido $G[n] = \frac{1}{\widetilde{R}[n]}$.

Definido o formato do limiar, o próximo passo é aplicar um ganho. Assim, os picos menos relevantes serão ignorados, reduzindo a quantidade de erros. A Figura 3.2 mostra o limiar gerado após a aplicação do SSE no fluxo espectral, exemplificado na Figura 3.1. Vale ressaltar que o ganho do limiar é um parâmetro de grande importância e que a escolha adequada de seu valor terá um impacto significante nos resultados [9]. No Capítulo 6, é descrito um método de obtenção do valor ótimo para esse parâmetro.

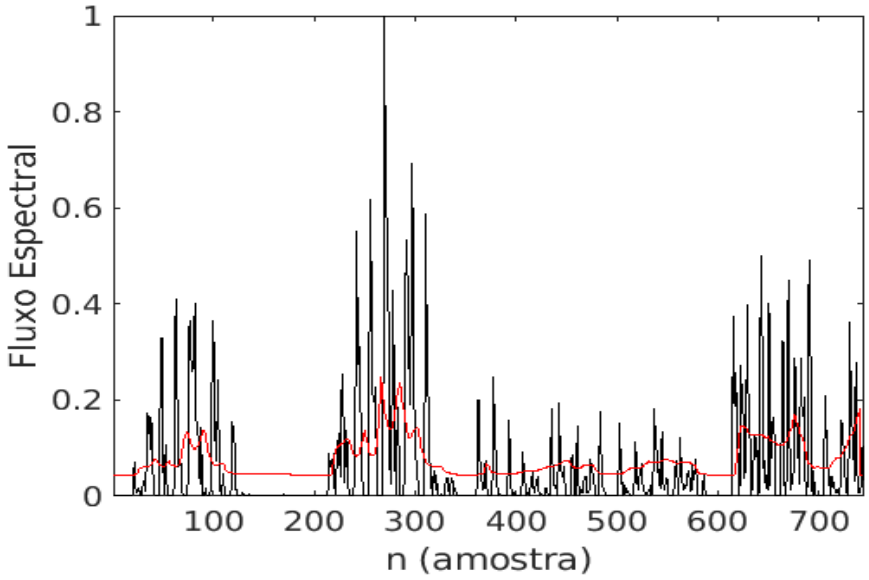

Figura 3.2: Fluxo Espectral e limiar de seleção de picos.

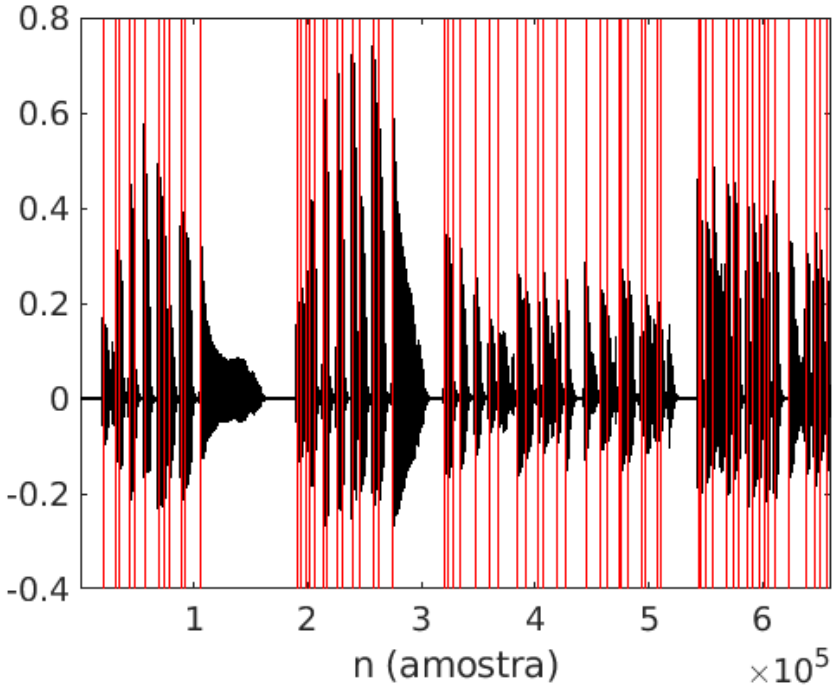

Figura 3.3: Sinal no tempo com *onsets* marcados.

Também é possível definir uma distância mínima entre picos adjacentes. Isso porque é altamente improvável a ocorrência de *onsets* consecutivos em um intervalo de tempo muito curto. Por fim, são armazenados os instantes associados aos picos selecionados. A Figura 3.3 mostra o resultado obtido para o sinal de exemplo descrito na Seção 2.1.

3.3 Avaliação

Para avaliar o desempenho do algoritmo de detecção de *onset*, é necessário, primeiramente, obter uma referência, que consiste em um conjunto de sinais com os *onsets* marcados. Esta pode ser uma etapa desafiadora, pois, na maioria dos casos, a marcação deve ser feita manualmente [9]. Isso torna todo o processo lento e susceptível a erros. Para amenizar esse problema, existem ferramentas que facilitam o procedimento, como a desenvolvida em [10]. Em relação aos erros, é interessante fazer uma validação dos *onsets* marcados por outra pessoa.

De posse de um conjunto de referência, é necessário estabelecer uma medida de desempenho. Geralmente, essa medida é expressa a partir de dois parâmetros: Precisão (P) e Revocação (R). Para calcular esses parâmetros, é necessário classificar os *onsets* em acertos, falsos positivos (*onsets* inexistentes detectados) e falsos negativos (*onsets*

existentes não detectados). A partir desses dados, podemos calcular tais parâmetros da seguinte maneira:

$$\begin{cases} P = \dfrac{c}{c + f^{+}} \\ R = \dfrac{c}{c + f}, \end{cases}$$

onde c, f^{+} e f são, respectivamente, o número de acertos, falsos positivos e falsos negativos [9]. Para uma avaliação mais geral, é desejado um balanço entre os dois parâmetros. Nesse caso, a medida mais utilizada é a Medida-F [9], que é calculada da seguinte forma:

$$F = \frac{2PR}{P + R} = \frac{2c}{2c + f^{+} + f}. \tag{3.1}$$

Dependendo do tipo de aplicação, a forma como esses parâmetros serão utilizados pode variar. Por exemplo, um algoritmo de estimação de andamento requer um valor alto de Precisão, mesmo que isso implique uma menor Revocação [8].

Capítulo 4

Rastreamento de frequência fundamental

No capítulo anterior, foi abordado o aspecto rítmico da música, que está relacionado com sua estrutura temporal. Outro aspecto essencial, que será discutido neste capítulo, é a melodia. Esta é geralmente o elemento mais memorável de uma peça musical, sendo, em muitos casos, o meio pelo qual o artista expressa suas ideias. Além de sua importância na arte, ela é utilizada em diversas aplicações, como o *query-by-humming*, que identifica uma peça musical a partir de sua versão cantarolada. Por isso, foram desenvolvidas diversas maneiras de se obter uma estimativa da melodia principal em uma música [3].

De forma geral, melodia pode ser definida como uma sucessão de notas musicais que, em conjunto, expressam uma ideia [1]. Em processamento de sinais de música, no entanto, costuma-se definir melodia de uma maneira diferente. Cada nota produzida por um instrumento musical possui, em sua composição espectral, uma estrutura harmônica, sendo uma componente a mais predominante. Essa componente é conhecida como frequência fundamental (f_0). No caso de sinais monofônicos, que serão os sinais analisados nesse capítulo, não há a ocorrência simultânea de diferentes notas. Isso significa que, para cada instante, existe no máximo uma f_0. Nessa situação, a melodia de uma música fica definida como a evolução temporal da frequência fundamental do sinal. Isso ocorre porque a f_0 está diretamente relacionada com a noção de *pitch*, que é a forma como o ser humano percebe os sons e os classifica como graves ou agudos [1]. Dessa forma, a tarefa de extração de melodia se resume a estimar a frequência fundamental do sinal ao longo do tempo. Esse procedimento é conhecido como Rastreamento de Frequência Fundamental.

Neste trabalho, o rastreamento será feito por meio da Função de Saliência, que parte da representação tempo-frequencial do sinal, abordada no Capítulo 2. Nas seções a seguir, serão descritas as principais etapas do processo, além de um método de avaliação dos resultados.

4.1 Seleção de picos

O método de rastreamento de f_0 utilizado consiste em avaliar as componentes frequenciais e atribuir um valor de saliência a cada uma. Realizar tal procedimento em todo o espectro do sinal implica um alto custo computacional. Por essa razão, é importante que se faça, primeiramente, uma redução no número de candidatos. Como foi visto na Seção 2.1, a janela de análise causa um espalhamento da energia das componentes espectrais do sinal em torno de seus valores originais. Isso significa que existem diversas componentes de baixa intensidade que podem ser interpretadas como ruído e, caso fossem consideradas pelo algoritmo, tornariam o número de candidatos muito grande. Como o método utilizado analisa cada candidato, o custo computacional do algoritmo aumentaria consideravelmente. Uma forma de amenizar esse problema é, para cada quadro de tempo, considerar apenas os picos mais relevantes presentes no espectro do sinal. Também é possível restringir os candidatos a um intervalo de frequências. Em um sinal de áudio, é pouco provável a ocorrência de notas cuja frequência fundamental se encontre abaixo de 20 Hz ou acima de 5000 Hz. As componentes com frequência fora desse intervalo podem, portanto, ser desconsideradas.

Além do intervalo de frequências, é utilizada a magnitude dos máximos locais como critério de seleção. No algoritmo proposto em [15], é definido um número fixo de picos selecionados por quadro do espectrograma (P). São selecionados então os P maiores picos no espectro e, por fim, os valores de amplitude e frequência são armazenados, formando assim um conjunto $\mathbf{P}_m = \mathbf{f}(f_0, a_0), (f_1, a_1), \ldots, (f_P \quad 1, a_P \quad 1), (f_P, a_P)\mathbf{g}$, onde m é o índice referente ao quadro do espectrograma. O conjunto $\mathbf{P}_m$ representa os picos do espectrograma que serão considerados como candidatos a f_0 na próxima seção.

Esse método de seleção, no entanto, não leva em consideração uma característica importante: sinais de áudio possuem um espectro do tipo passa-baixas [17]. Isso significa que as frequências mais altas têm magnitudes mais baixas, o que as deixam em

desvantagem na seleção de picos. Para resolver esse problema, será utilizado um pré-processamento, como sugerido em [16], utilizando o SSE, apresentado na Seção 3.2. Ao dividirmos o espectro original pelo filtrado, temos, como resultado um espectro planificado. Na Figura 4.1, temos o espectro de um trecho do sinal de exemplo apresentado na Seção 2.1.

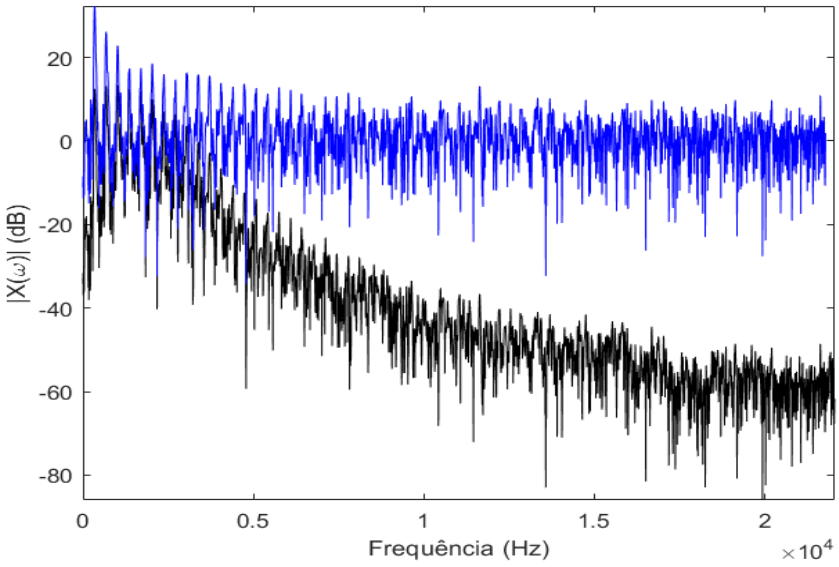

Figura 4.1: Espectro do sinal antes do processamento (preto/inferior) e depois do processamento (azul/superior).

4.2 Função de saliência

Saliência é uma medida de relevância de cada componente espectral em um determinado quadro de tempo e é o principal parâmetro utilizado no rastreamento de f_0. Por isso, uma etapa fundamental é o cálculo dessa medida ao longo do tempo, conhecida como Função de Saliência [13]. A forma mais comum de obter tal função é a partir da soma ponderada das amplitudes de cada componente e de seus respectivos harmônicos [14]. Esse cálculo é realizado para todos os candidatos a f_0 em cada quadro de tempo m, de acordo com a seguinte expressão:

$$S[m,k] = \sum_{h=1}^{H} w_h \, \boldsymbol{j}X[m,hk]\boldsymbol{j}\,,$$

onde $\boldsymbol{j}X[m,k]\boldsymbol{j}$ é o módulo do espectrograma do sinal, h é o índice referente ao harmônico da componente k, H é o número de harmônicos a serem considerados e w_h é um peso

atribuído ao candidato.

Uma característica desse método é a sua dependência em relação ao timbre do instrumento, já que este está altamente relacionado com a energia presente nos harmônicos. Por isso, foi proposta em [13] uma maneira alternativa de calcular a saliência, mais robusta a variações de timbre. Em vez de utilizar a amplitude das componentes, o método considera a localização em frequência dos candidatos e seus respectivos harmônicos. De forma mais específica, é calculada a diferença entre a frequência dos harmônicos de um dado candidato e a frequência das notas na escala igualmente temperada. As frequências dessa escala podem ser obtidas a partir da seguinte equação:

$$f_i = f_{\mathsf{ref}} 2^{\frac{i \quad D}{12}}, \tag{4.1}$$

onde i *2* N é o índice MIDI (*Musical Instrument Digital Interface*) [13] referente à nota musical, f_{ref} é a frequência de referência (geralmente $f_{\mathsf{ref}} = 440\text{Hz}$) e f_i é a frequência da nota. O deslocamento de $D = 69$ é feito para incluir as notas que possuem frequência fundamental menor do que f_{ref}.

A Figura 4.2 mostra um exemplo com as notas da escala, partindo da nota $C4$ ($i = 60, f_{60} = 261,6\text{Hz}$). É possível perceber que a localização dos harmônicos (representada em linha tracejada) não está sempre alinhada com as respectivas notas da escala (em linha contínua), o que é mais evidente no quinto e no sétimo harmônico. Esses desvios formam uma sequência, que será utilizada como referência para calcular a saliência dos picos selecionados. Na proposta de [13], a unidade usada para medir os desvios é o *cent*, que é definido como um centésimo da distância entre duas notas consecutivas na escala igualmente temperada. Como uma oitava contém 12 notas, temos um total de 1200 *cents* por oitava.

Para uma fundamental f_0, a frequência de seu h-ésimo harmônico é dada como:

$$f_h^{f_0} = h \quad f_0, \tag{4.2}$$

onde $h = 2, 3, \ldots, H$. O desvio, em *cents*, do harmônico $f_h^{f_0}$ em relação à escala de igual temperamento é definido da seguinte maneira:

$$d_h^{f_0} = 100 \left[12log_2 \left(\frac{f_h^{f_0}}{f_{\mathsf{ref}}} \right) \quad \left\lfloor 12log_2 \left(\frac{f_h^{f_0}}{f_{\mathsf{ref}}} \right) \right\rceil \right], \tag{4.3}$$

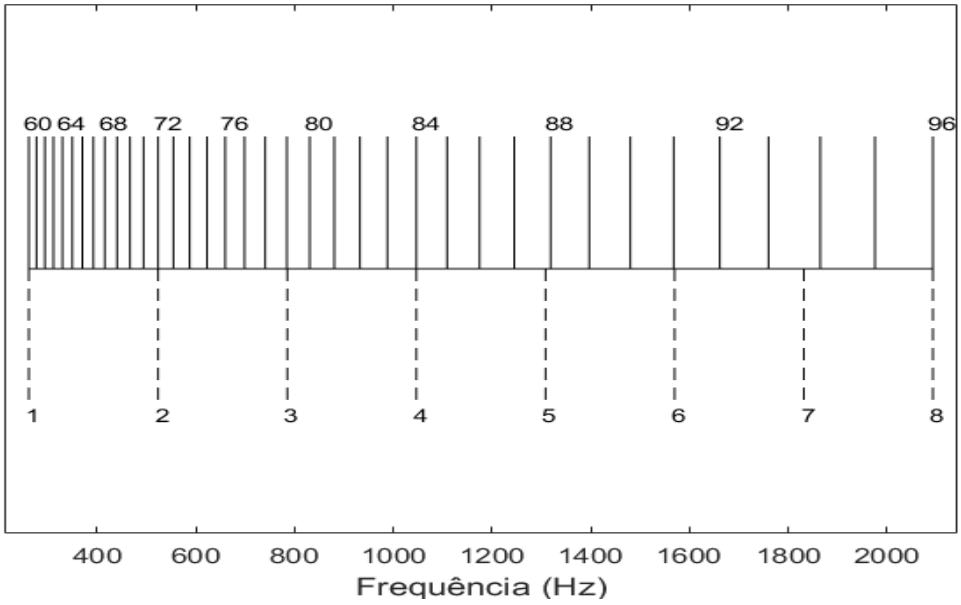

Figura 4.2: Localização frequencial das notas da escala igualmente temperada (linha contínua) e dos harmônicos da nota C4 (261,6 Hz) (linha tracejada).

onde $\lfloor \cdot \rceil$ é a operação de arredondamento. Substituindo as Equações (4.1) (sem o deslocamento de 69, por simplicidade) e (4.2) em (4.3), obtemos:

$$d_h^{f_i} = 100[12log_2(h) \quad \lfloor 12log_2(h) \rceil]. \tag{4.4}$$

A Equação (4.4) deixa claro que o desvio não depende do valor de f_0 [13]. Por essa razão, simplificamos a sua notação para d_h. Definimos como $\{d_h\}$ a sequência de desvios teóricos, que será comparada com os desvios observados $\{\hat{d}_h^{f_p}\}$. Estes são computados para cada pico $p \in P_m$.

Para encontrarmos os desvios de um dado candidato com frequência f_p, buscamos, dentre os demais picos $f_{p'}$, quais correspondem aos harmônicos de f_p. Isso é feito por meio de uma função gaussiana G centrada no h-ésimo harmônico de f_p, avaliada em $f_{p'}$. Especificamente, temos $G[f_{p'}; \mu_{h,p}, \sigma_{h,p}]$, onde a média é dada por $\mu_{h,p} = hf_p$ e o desvio padrão é $\sigma_{h,p} = \mu_{h,p}\left(2^{\frac{\alpha}{1200}} \quad 1\right)$. O parâmetro $\alpha = 20$ foi escolhido experimentalmente para compensar erros gerados durante a etapa de seleção de picos [13].

A função gaussiana é utilizada por possuir valor máximo quando $f_{p'} = \mu_{h,p}$ e tender a zero para os demais valores de $f_{p'}$. Assim, ao selecionarmos o maior valor de G é possível identificar qual pico corresponde ao harmônico $f_h^{f_p}$. Cada pico $f_{p'}$ é então associado a um desvio $d_{p'}$, que é calculado pela Equação (4.3), como segue:

$$d_{p^0} = 100 \left[12log_2 \left(\frac{f_{p^0}}{f_{\mathsf{ref}}} \right) \quad \left\lfloor 12log_2 \left(\frac{f_{p^0}}{f_{\mathsf{ref}}} \right) \right\rfloor \right].$$

Por fim, o desvio $\hat{d}_h^{f_p}$ é calculado como uma soma ponderada dos desvios d_{p^0}:

$$\hat{d}_h^{f_p} = \sum_{p^0=1}^{P} G[f_{p^0}; \mu_{h,p}, \sigma_{h,p}] \quad d_{p^0}. \tag{4.5}$$

Os valores obtidos de $\hat{d}_h^{f_p}$ formam a sequência $\mathsf{f}\hat{d}_h^{f_p}\mathsf{g}$, que será usada para encontrarmos a saliência do pico f_p.

A saliência de um pico, denotada por $S(p)$, é então calculada como a correlação entre as sequências $\mathsf{f}d_h\mathsf{g}$ e $\mathsf{f}\hat{d}_h^{f_p}\mathsf{g}$. Em [13], a medida utilizada é o produto interno. Aqui, será utilizado o coeficiente de correlação, proposto em [16]. Primeiramente, escrevemos as sequências na forma vetorial $\mathbf{d}^{f_p} = [d_2^{f_p} d_3^{f_p} \ldots d_H^{f_p}]^T$ e $\hat{\mathbf{d}}^{f_p} = [\hat{d}_2^{f_p} \hat{d}_3^{f_p} \ldots \hat{d}_H^{f_p}]^T$. O coeficiente de correlação $r(p)$ é definido como:

$$r[p] = \frac{\mathsf{h}\mathbf{d}^{f_p} \quad \boldsymbol{\mu}_{\mathsf{teo}}, \hat{\mathbf{d}}^{f_p} \quad \boldsymbol{\mu}_{\mathsf{obs}}\mathsf{i}}{\sqrt{\sigma_{\mathsf{teo}}^2}\sqrt{\sigma_{\mathsf{obs}}^2}},$$

onde μ_{teo} e μ_{obs} são, respectivamente, as médias dos vetores $\mathbf{d}^{f_p}$ e $\hat{\mathbf{d}}^{f_p}$; os vetores $\boldsymbol{\mu}_{\mathsf{teo}}$ e $\boldsymbol{\mu}_{\mathsf{obs}}$ possuem $H \quad 1$ elementos, com valores iguais a μ_{teo} e μ_{obs}; σ_{teo}^2 e σ_{obs}^2 são, respectivamente, as variâncias de $\mathbf{d}^{f_p}$ e $\hat{\mathbf{d}}^{f_p}$ e $\mathsf{h}\ ,\ \mathsf{i}$ é a operação de produto interno. Para reduzir a influência de componentes de baixa amplitude, ponderamos o coeficiente de correlação pela amplitude do pico a_p. Por fim, a saliência é calculada da seguinte maneira:

$$S[p] = a_p r[p]. \tag{4.6}$$

O cálculo de saliência, dado pela Equação (4.6), é realizado para cada candidato a f_0. Em cada quadro de tempo, é escolhido o candidato que possui o maior valor de saliência. As componentes selecionadas formam, por fim, uma sequência $\mathsf{F}[m] = \mathsf{f}\hat{f}_0^m\mathsf{g}$, que associa a cada quadro m o valor da frequência fundamental $\hat{f}_0^m$ computada.

O rastreamento de f_0 pelo método de saliência independente do timbre possui uma característica que deve ser levada em consideração ao analisarmos o sinal. Por utilizar a escala de igual temperamento, esse algoritmo depende da afinação correta dos instrumentos, para que a Equação (4.4) se mantenha válida. Caso o instrumento esteja desafinado, ou afinado de acordo com outra escala, a sequência de desvios observados possuirá uma baixa correlação com a sequência de referência, mesmo para os picos que, em tese, correspondem à f_0. No entanto, a função gaussiana, utilizada na obtenção dos harmônicos

dos candidatos, permite uma certa tolerância. Isso porque frequências próximas do centro dessa função ainda produzem valores aceitáveis de G, não afetando tanto o cálculo dos desvios pela Equação (4.5). Porém, caso a diferença entre as frequências das notas produzidas e as frequências da escala de igual temperamento seja maior do que o parâmetro α do desvio padrão de G, os resultados serão consideravelmente penalizados [13].

4.3 Avaliação

Para cada quadro de tempo, o método de rastreamento de f_0 utilizado retorna um valor que corresponde à frequência fundamental do sinal naquele instante. Para determinarmos se os valores encontrados estão corretos, é necessário utilizar um conjunto de referência. Neste trabalho, será usada como referência a base de dados "MDB-mf0-synth" [18], que contém sinais retirados da base "MedleyDB" [19], além das anotações da frequência fundamental para cada sinal. A base "MDB-mf0-synth" foi escolhida porque os sinais nela contidos foram modificados, de forma que os instrumentos polifônicos foram removidos. Isso significa que cada instrumento presente no sinal produz uma única frequência fundamental por vez, facilitando assim a comparação com os resultados obtidos. Além disso, os instrumentos monofônicos foram ressintetizados, para melhorar a confiabilidade das anotações [18].

Para avaliar a sequência $\mathsf{F}[m]$, é calculada, quadro a quadro, a distância euclidiana unidimensional entre as frequências computadas $\hat{f}_0^m$ e as frequências da base de dados f_0^m, como segue:

$$D = \mathsf{j}\hat{f}_0^m \quad f_0^m\mathsf{j}. \tag{4.7}$$

É definido também um valor de tolerância T. Dentre os elementos de $\mathsf{F}[m]$, serão considerados como acertos aqueles que satisfazem a relação $D \quad T$. Por fim, é calculada a precisão dos resultados obtidos, da seguinte maneira:

$$P = \frac{c}{c+e}, \tag{4.8}$$

onde c e e são, respectivamente, o número de acertos e de erros.

Uma característica notável desse algoritmo de rastreamento de f_0 é o fato de que todos os quadros são associados a alguma f_0. Porém, em uma peça musical, é comum a existência de períodos de silêncio, ou seja, não há uma frequência fundamental a ser a

associada nesses instantes. Isso leva a um aumento no número de erros, reduzindo assim a precisão do algoritmo. Também deve ser notado o fato de que cada quadro é associado a um único valor de f_0, o que deixa de levar em consideração a ocorrência de múltiplas notas sendo tocadas simultaneamente. No Capítulo 7, serão abordadas propostas para contornar esses problemas.

Capítulo 5

Separação de componentes tonais e transitórias

Foi visto no Capítulo 4 que uma nota musical pode ser descrita, no domínio da frequência, a partir de uma estrutura formada por uma frequência fundamental e seus harmônicos. No entanto, existem outras características que não podem ser descritas ao considerarmos o som apenas como a superposição de tais componentes. Uma nota musical, ao ser tocada, possui atributos que mudam constantemente ao longo do tempo [1]. Portanto, uma forma mais acurada de representar esse evento musical também deve levar em consideração a sua progressão temporal.

Uma maneira simples, porém interessante, de descrever o comportamento do som ao longo do tempo é o modelo ADSR (Ataque, Decaimento, Sustentação, Relaxamento), que é muito utilizado por sintetizadores para reproduzir sons de instrumentos reais. Esse modelo divide a nota musical em quatro etapas, de acordo com a amplitude do sinal no tempo de duração da nota [1]. A Figura 5.1 mostra uma representação gráfica das etapas desse modelo. Na primeira etapa, conhecida como fase de ataque, a amplitude do sinal cresce rapidamente. Durante essa fase, há o surgimento de componentes ao longo de todo o espectro. Após o ataque, é iniciada a fase de decaimento, na qual a amplitude do sinal decai durante um intervalo curto de tempo, até estabilizar. A etapa seguinte, chamada de fase de sustentação, costuma ser a etapa de maior duração da nota. Nela, o som se torna aproximadamente estacionário e sua amplitude se mantém constante ou decai lentamente. Por fim, ocorre a fase de relaxamento, marcando o fim da nota [1].

É importante notar que nem todos os instrumentos podem ser corretamente des-

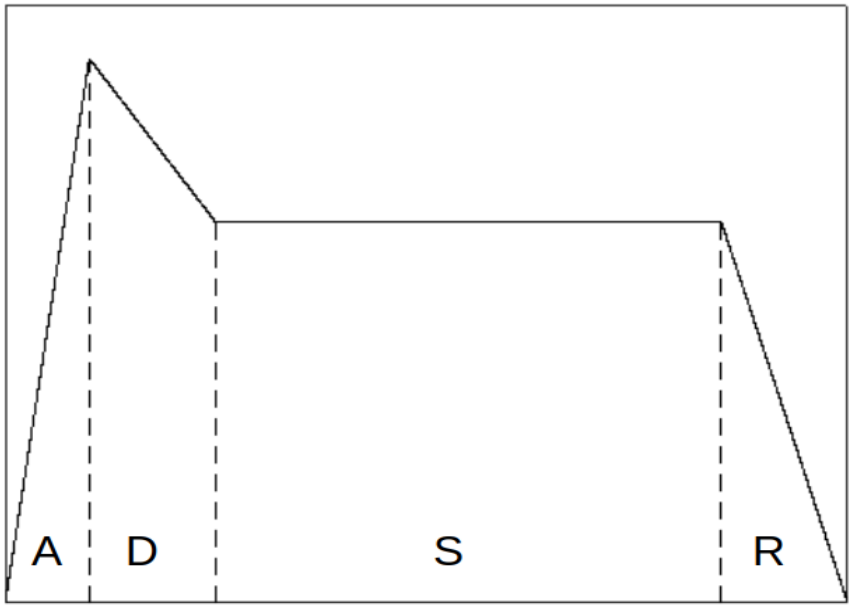

Figura 5.1: Etapas do modelo ADSR.

critos por esse modelo. As notas produzidas por um violino, por exemplo, podem possuir uma fase de ataque longa, em comparação com outros instrumentos. Além disso, pode nem haver uma fase de decaimento, já que o intérprete pode controlar o som de forma a aumentar a intensidade da nota até ocorrer o relaxamento.

Apesar de ser uma simplificação, o modelo ADSR nos permite descrever uma nota musical de maneira mais precisa, levando em consideração as mudanças em seus atributos ao longo do tempo. Além disso, podemos identificar a manifestação de características distintas da nota em diferentes instantes. Durante as fases de ataque e decaimento, podemos observar que o sinal apresenta características aproximadamente impulsivas, marcadas pelo surgimento de componentes ao longo de todo o espectro, seguido rapidamente pelo desaparecimento dessas mesmas componentes. Esse fenômeno predomina nas emissões dos instrumentos de percussão e pode ser chamado de componente transitória de uma nota musical. A fase de sustentação, por outro lado, nos permite observar, no espectrograma, a estrutura harmônica do sinal. Durante esse intervalo, o som é percebido como uma superposição de tons. Essa característica é então chamada de componente tonal da nota musical.

A identificação de tais componentes é particularmente interessante para a extração de certas informações do sinal, como, por exemplo, os *onsets* e a f_0. O algoritmo de detecção de *onset*, visto no Capítulo 3, busca os instantes em que há um aumento súbito

na energia das componentes espectrais. Esses instantes correspondem à fase de ataque das notas musicais. Isso significa que as informações relevantes para a detecção de *onset* estão contidas na componente transitória dos eventos. Consequentemente, a presença da componente tonal é desnecessária, podendo causar erros na detecção. De maneira análoga, temos que o algoritmo de rastreamento de f_0 utiliza a componente tonal das notas, enquanto as propriedades transitórias podem ser problemáticas para a determinação da melodia principal da música [1].

Assim sendo, a separação das componentes tonais e transitórias deve reduzir a quantidade de erros em ambos os casos, melhorando o desempenho dos algoritmos. Na seção a seguir, será descrito o método utilizado para realizar essa separação.

5.1 Filtragem por mediana

A separação das componentes pode ser feita de diversas maneiras [20, 21]. No entanto, a maioria dos algoritmos requer um grande conjunto de dados, por utilizarem técnicas de aprendizado de máquina, ou requerem uma grande quantidade de recursos computacionais, como capacidade de processamento e armazenamento em memória [22]. Por essa razão, esses métodos podem não ser boas opções para processar músicas inteiras. Diante dessa situação, foi proposto em [22] um algoritmo de separação que parte da ideia de que as componentes tonais e transitórias formam, no espectrograma do sinal, linhas horizontais e verticais, respectivamente [23]. A Figura 5.2 mostra o espectrograma de um sinal gerado por um piano solo, amostrado a uma taxa de $44,1$ kHz. Nela, é possível perceber as linhas verticais, que ocorrem no instante em que as notas são tocadas, e as linhas horizontais, que correspondem à fase de sustentação de cada nota. Dessa forma, o desafio de separar tais componentes passa a ser entendido como uma tarefa de retirar do espectrograma as linhas indesejadas. O método proposto em [22] para realizar essa tarefa utiliza a filtragem por mediana móvel. Esse filtro geralmente é utilizado em processamento de imagens para remover um tipo de ruído conhecido como “sal e pimenta” [24].

De forma empírica, a mediana corresponde ao valor central de um conjunto finito de números reais, após organizar seus elementos em ordem crescente. Caso o número de elementos seja par, a mediana é definida como a média entre os dois valores centrais do conjunto. Ao selecionarmos um trecho de um sinal, podemos aplicar a mediana para gerar

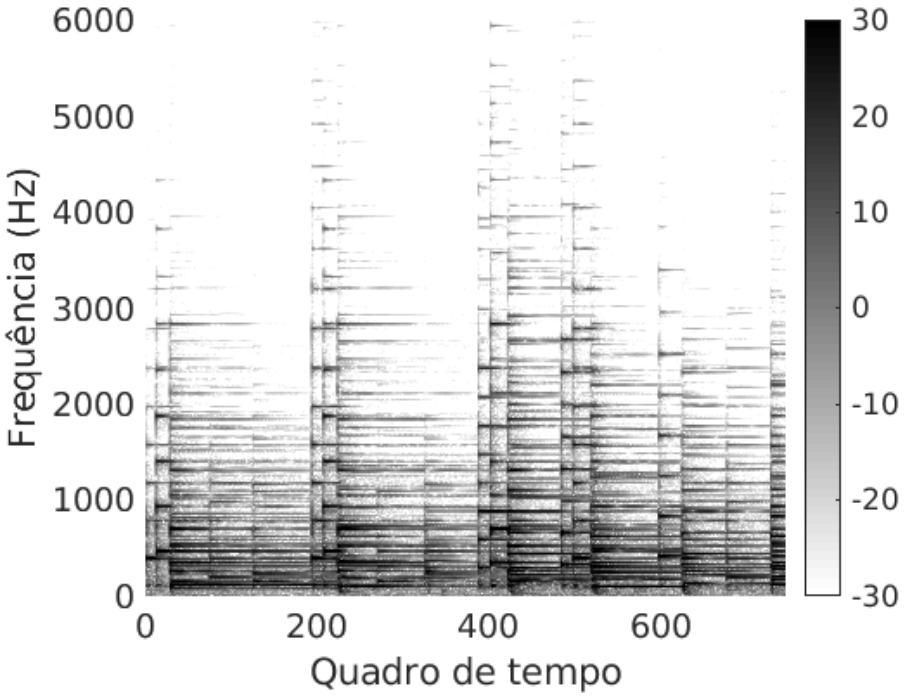

Figura 5.2: Espectrograma de um sinal gerado por um piano.

o filtro. O processo de filtragem do sinal ocorre ao substituirmos uma dada amostra pela mediana local, considerando uma região em torno dessa amostra. Em [22], é aplicada uma versão unidimensional do filtro. Dessa forma, a região a ser considerada é um intervalo de comprimento L, em amostras. Por simplicidade, assumimos que L é um número ímpar. O filtro de mediana fica então definido da seguinte maneira, para um sinal $x[n]$:

$$y[n] = \text{mediana}\boldsymbol{f}x[n \quad k : n + k]\boldsymbol{g},$$

onde $y[n]$ é o sinal filtrado e $k = \dfrac{L \quad 1}{2}$. Esse processo é repetido para cada valor de $x[n]$.

A principal característica desse filtro é a sua capacidade de remover picos do sinal, sem causar grandes alterações na região em torno desses pontos. Tal característica é o que torna essa técnica apropriada para a separação das componentes, ao ser aplicada no módulo do espectrograma de um sinal, aqui denotado por $\mathbf{S}$. É importante notar que o tamanho do intervalo considerado na mediana impacta diretamente no resultado da separação. Quanto maior o valor de L, maior o efeito de supressão dos picos. Além disso, um aumento no comprimento do filtro implica um número maior de amostras a serem consideradas no cálculo da mediana, o que aumenta o custo computacional.

Ao analisarmos uma dada linha espectral ao longo do tempo, temos que os eventos transitórios estão associados a picos nos seus respectivos segmentos de ocorrência.

Analogamente, ao analisarmos as componentes presentes uma dada janela, temos que as componentes f_0 e seus harmônicos correspondem a picos no espectro [22]. Dessa forma, caso o filtro de mediana seja aplicado na horizontal, sua componente transitória será atenuada. Ao realizarmos a filtragem em cada linha de $\mathbf{S}$, temos um novo espectrograma $\mathbf{H}$, que possui sua componente tonal enfatizada. De maneira similar, podemos gerar uma representação com componentes tonais atenuadas ao aplicarmos o filtro na vertical, em cada coluna de $\mathbf{S}$, resultando em um espectrograma $\mathbf{T}$ com sua componente transitória enfatizada.

Os dois espectrogramas filtrados $\mathbf{H}$ e $\mathbf{T}$, no entanto, não podem ser diretamente utilizados para construir as componentes tonais e transitórias [1]. É necessário gerar máscaras, que serão aplicadas ao espectrograma original. Em [22], é sugerido o uso de máscaras baseadas no filtro de Wiener, como segue:

$$\mathbf{M_H} = \mathbf{H}^p./(\mathbf{H}^p + \mathbf{T}^p)$$

$$\mathbf{M_T} = \mathbf{T}^p./(\mathbf{H}^p + \mathbf{T}^p),$$

onde cada elemento de $\mathbf{H}$ e $\mathbf{T}$ é elevado à potência p e o símbolo $(./)$ representa uma divisão ponto a ponto. Valores comuns para p são 1 ou 2. Por fim, cada elemento do espectrograma original $\hat{\mathbf{S}}$, com valores complexos, é multiplicado por uma das máscaras:

$$\hat{\mathbf{H}} = \hat{\mathbf{S}} \quad \mathbf{M_H}$$

$$\hat{\mathbf{T}} = \hat{\mathbf{S}} \quad \mathbf{M_T}.$$

O símbolo representa a multiplicação ponto a ponto. Os resultados obtidos, como

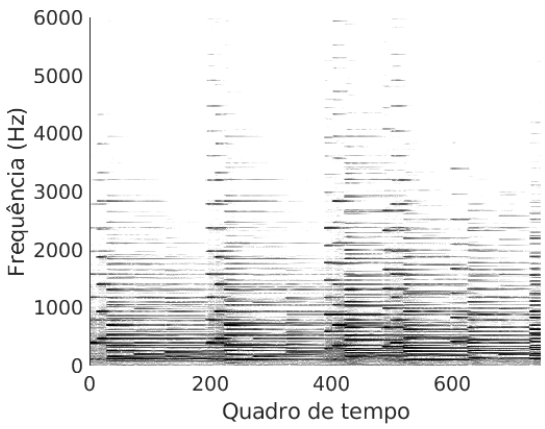

Figura 5.3: Componente tonal.

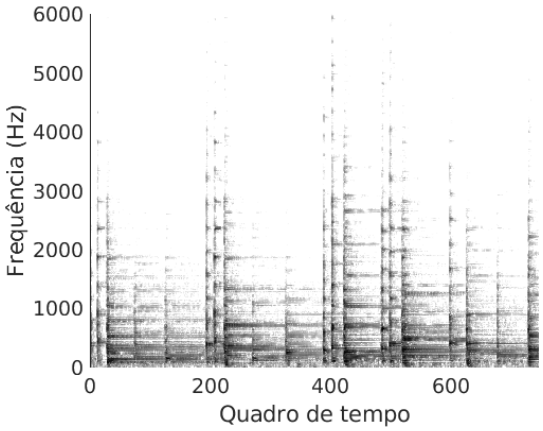

Figura 5.4: Componente transitória.

mostram as Figuras 5.3 e 5.4, são os espectrogramas, cada um com uma componente enfatizada e a outra atenuada. Esses novos espectrogramas serão então utilizados como entrada nos algoritmos descritos nos Capítulos 3 e 4. No Capítulo 6, é realizada a comparação do desempenho dos algoritmos, a fim de verificar a eficácia da separação das componentes.

Capítulo 6

Experimentos e resultados

Com o objetivo de avaliar a influência da separação de componentes no desempenho dos algoritmos descritos nos capítulos anteriores, foram realizados alguns experimentos no *software* Matlab. Primeiramente, foram definidos os parâmetros básicos, como, por exemplo, o tamanho e o salto da janela de análise para gerar o espectrograma. Depois, os algoritmos de detecção de *onset* e rastreamento de f_0 foram executados e os resultados obtidos foram armazenados. Por fim, foi adicionada a etapa de pré-processamento descrita no Capítulo 5 e os experimentos foram realizados novamente, mantendo os outros parâmetros previamente definidos. Todos os sinais analisados foram amostrados a uma taxa $F_s = 44,1$kHz e o espectrograma foi gerado pela STFT, por facilidade de implementação e baixo custo computacional. Por fim, foi estabelecido que um aumento de 10% nas métricas de avaliação era suficiente para considerarmos o resultado aceitável.

6.1 Detecção de *onset*

Como foi visto na Seção 3.3, o conjunto de referência é gerado a partir das marcações manuais dos *onsets*. Além disso, é importante, para garantir a confiabilidade das anotações, que os *onsets* marcados sejam validados por outras pessoas. Por essa razão, foi utilizada a base de dados[1] criado em [10], que consiste em 16 sinais gerados por instrumentos solo, cada um com duração de aproximadamente 15 segundos. Os *onsets* desses sinais foram anotados por três pessoas diferentes e os resultados foram comparados. Por fim, foram armazenadas em um conjunto as anotações consistentes, ou seja, aquelas sobre

[1] https://adasp.telecom-paris.fr/resources/2011-07-13-sound-onset-labellizer/

as quais houve uma concordância entre as três pessoas. No total, foram registrados 678 *onsets* consistentes.

Vale notar que, mesmo com a validação por outras pessoas, a marcação manual não é capaz de estabelecer com exatidão o instante de ocorrência de cada evento musical. Por essa razão, é comum definir um intervalo de tolerância para que o *onset* computado seja considerado um acerto. Aqui, o *onset* será considerado como acerto caso a distância entre o valor encontrado e a referência esteja dentro de uma janela de 50 ms para cada lado, como sugerido em [9].

Para a realização dos experimentos, foram definidos, em primeiro lugar, os parâmetros da STFT. Tais parâmetros foram definidos a partir de testes, utilizando alguns valores sugeridos em [12]. Como janela de análise $w[n]$, foi utilizada uma janela de Hamming de 75 ms (3308 amostras), com um salto temporal $h = 20$ ms (882 amostras). Após a obtenção do espectrograma, foi gerado o fluxo espectral, que teve sua faixa dinâmica comprimida para o intervalo de 0 a 1. Depois, foi obtido o limiar, utilizando o SSE com um filtro de média móvel de 30 amostras de comprimento. Como foi visto na Seção 3.2, deve ser aplicado um ganho no limiar, para então selecionar os picos. O valor desse ganho foi determinado, para cada sinal, por meio de uma busca exaustiva, considerando um conjunto de valores dentro do intervalo de 0 a 1. Essa busca é feita para aferir o melhor desempenho alcançável, já que, como foi visto na Seção 3.2, o ganho do limiar tem grande influência nos resultados obtidos pelo algoritmo.

Para cada ganho aplicado, foram selecionados os picos acima do limiar. Também foi definida uma distância mínima de 33 ms entre picos adjacentes. Os picos selecionados, que correspondem aos *onsets* computados, foram então comparados com o conjunto de referência e a Medida-F foi calculada, segundo a Equação (3.1). Dentre os valores de ganho considerados, foi selecionado aquele que estava associado ao maior valor de F. Os valores calculados foram, por fim, armazenados. A Tabela 6.1 mostra o resultado obtido para cada sinal da base de dados, bem como a média dos valores encontrados.

De posse de uma avaliação inicial do desempenho do algoritmo, foi verificada a influência da separação das componentes tonais e transitórias. Para isso, o comprimento do intervalo considerado no filtro de mediana foi definido como $L \in \{10, 20, 30\}$ amostras. Para cada valor de L, foi realizada a separação e o espectrograma com componentes transitórias enfatizadas $\hat{\mathbf{T}}$ foi utilizado para obtermos o novo conjunto de *onsets*.

Tabela 6.1: Medida-F para cada sinal da base de dados.

#	Sinal	F
1	guitar3	0,966
2	synthbass1	0,958
3	distguit1	0,807
4	trumpet1	0,778
5	rock1	0,671
6	guitar2	0,667
7	techno2	0,647
8	jazz3	0,644
9	piano1	0,640
10	violin2	0,639
11	cello1	0,615
12	classic2	0,614
13	pop1	0,530
14	jazz2	0,493
15	sax1	0,333
16	clarinet1	0,077
Média		0,630

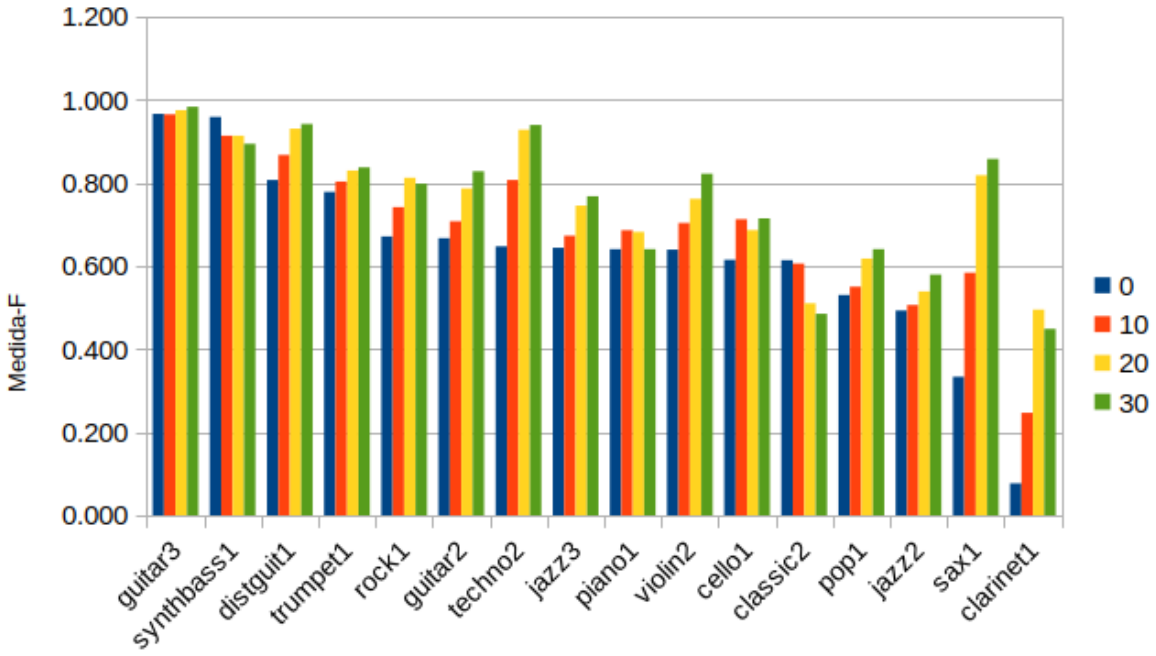

Figura 6.1: Medida-F para cada valor de L. Os valores associados a $L = 0$ correspondem aos resultados obtidos sem o pré-processamento.

A Figura 6.1 mostra os resultados obtidos para cada sinal. As médias dos valores encontrados foram, respectivamente, iguais a $f0,692;\ 0,752;\ 0,761g$, o que representa um aumento de $f9,83\%;\ 19,35\%;\ 20,79\%g$ em relação ao valor obtido sem o pré-processamento. Como o valor mínimo para esse aumento foi estabelecido como 10%, temos que um filtro de comprimento $L = 10$ amostras não produziu resultados aceitáveis. Para os valores de $L\ 2\ f20,30g$, podemos observar que a separação das componentes resultou em um aumento interessante no valor de F. Em especial, os sinais gerados por instrumentos de sopro, como o clarinete e o saxofone, foram muito beneficiados pela filtragem. Além disso, valores maiores de L levaram a um resultado melhor, na maioria dos casos. Porém, é importante notar que, quanto maior o valor de L, maior o custo computacional do algoritmo, como visto na Seção 5.1. Considerando tal fato, temos que um filtro de comprimento $L = 30$ amostras pode não ser interessante, já que este não produziu resultados muito superiores em relação ao filtro de $L = 20$ amostras.

6.2 Rastreamento de frequência fundamental

Para avaliar o algoritmo de rastreamento de f_0, foram utilizados 15 sinais da base de dados "MDB-mf0-synth" [18]. Como explicado na Seção 4.3, os sinais contidos nessa base foram modificados, de forma que os instrumentos polifônicos foram suprimidos e o som gerado pelos instrumentos monofônicos foi ressintetizado. Porém, nesses sinais, existem trechos nos quais múltiplos instrumentos são tocados simultaneamente. Isso resulta em uma polifonia, ou seja, há mais de uma frequência fundamental nesses intervalos. Como o algoritmo utilizado considera o sinal monofônico, os trechos onde ocorre polifonia não foram considerados na avaliação.

Para a realização do experimento, foram definidos, primeiramente, os parâmetros da STFT. O tamanho da janela análise foi definido por meio de testes. A janela escolhida foi uma janela de Hamming de 100 ms (4410 amostras), com um salto temporal $h = 10$ ms (441 amostras), que é o valor utilizado nas anotações do conjunto de referência. A partir do espectrograma gerado, foi feita a seleção de picos, resultando no conjunto P_m. Para essa etapa, foi definido o número de picos por quadro $P = 80$, como sugerido em [16]. Depois, foi definido o número de harmônicos a serem considerados para a obtenção da sequência de desvios. O valor utilizado foi $H = 8$. Por fim, a frequência de referência foi definida como $f_{\text{ref}} = 440$ Hz.

A partir dos valores computados, foi realizada a comparação com as anotações da base, utilizando a Equação (4.7). Como tolerância, foi definido um intervalo de $T = 50$ Hz para cada lado. A precisão dos resultados, calculada pela Equação (4.8), pode ser observada na Tabela 6.2.

De maneira similar àquela descrita na Seção 6.1, o algoritmo de rastreamento de f_0 foi realizado novamente, utilizando o espectrograma com componentes tonais enfatizadas $\hat{\mathbf{T}}$, considerando $L \in \{10, 20, 30\}$. Seguem os resultados na Figura 6.2.

Os valores médios dos resultados, para cada comprimento de filtro são, respectivamente, iguais a $\{0,566;\ 0,590;\ 0,599\}$, ou seja, houve um aumento de $\{7,04\%;\ 11,43\%;\ 13,18\%\}$ na Precisão. Nesse caso, é possível perceber que a separação não foi tão impactante nos resultados, em comparação com o algoritmo de detecção de *onset*. O filtro de comprimento $L = 10$ amostras produziu resultados abaixo do valor mínimo aceitável. Porém, houve uma melhora nos valores obtidos, na maior parte dos sinais analisados. Além disso, temos que, assim como na Seção 6.1, os resultados tendem a melhorar para

Tabela 6.2: Precisão para cada sinal da base de dados.

#	Sinal	Precisão
1	MusicDelta_ChineseDrama	0,671
2	StevenClark_Bounty	0,669
3	MusicDelta_Country1	0,667
4	MusicDelta_FusionJazz	0,607
5	MusicDelta_SwingJazz	0,596
6	Auctioneer_OurFutureFaces	0,565
7	MusicDelta_80sRock	0,541
8	MusicDelta_Pachelbel	0,523
9	MusicDelta_ChineseChaoZhou	0,505
10	AvaLuna_Waterduct	0,486
11	MusicDelta_BebopJazz	0,480
12	MatthewEntwistle_DontYouEver	0,438
13	MusicDelta_ChineseHenan	0,430
14	AlexanderRoss_VelvetCurtain	0,388
15	StrandOfOaks_Spacestation	0,371
Média		0,529

maiores valores de L. Contudo, considerando o aumento no custo computacional, o filtro de mediana com comprimento igual a 30 amostras é menos efetivo.

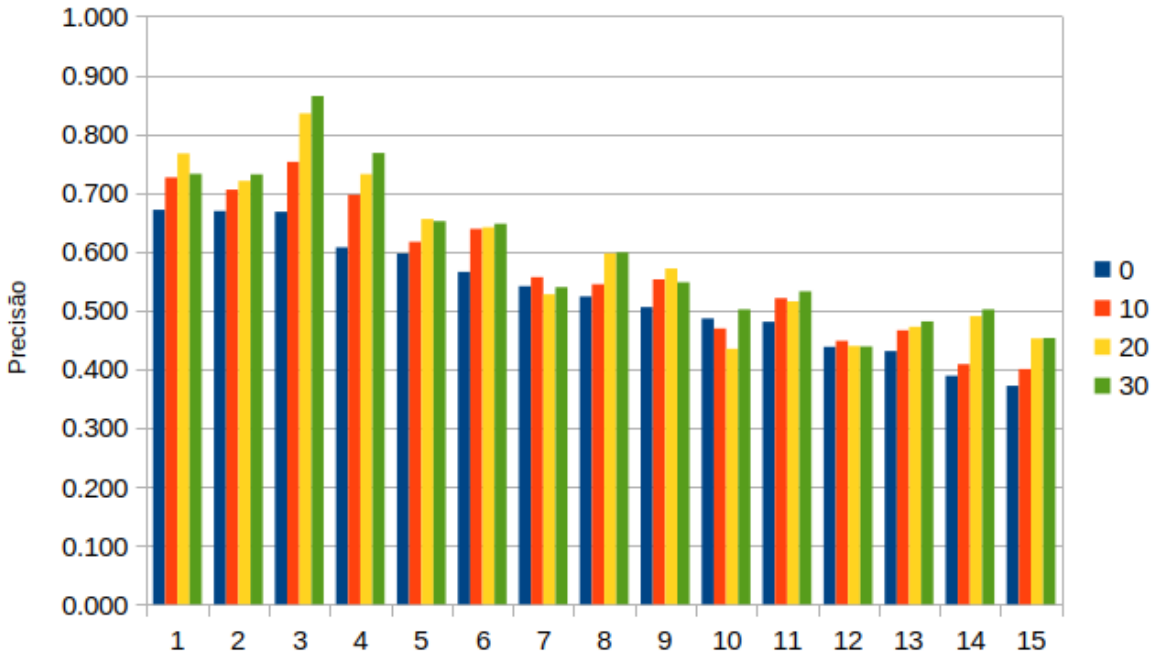

Figura 6.2: Precisão para cada valor de L. Os valores associados a $L = 0$ correspondem aos resultados obtidos sem o pré-processamento.

Capítulo 7

Conclusão e trabalhos futuros

7.1 Conclusão

Neste trabalho, foram estudadas algumas técnicas utilizadas na área de "Extração de Informação Musical". No Capítulo 2, foi apresentada uma forma de representar um sinal que torna facilmente observáveis as variações em sua composição espectral ao longo do tempo. Foram descritas, nesse capítulo, três maneiras diferentes de gerarmos essa representação, cada uma com suas vantagens e desvantagens.

Em seguida, no Capítulo 3, foi mostrado um algoritmo amplamente utilizado para a detecção dos instantes de ocorrência de eventos musicais [9]. Além disso, foram exploradas algumas modificações no método utilizado, como a utilização do SSE para a obtenção do limiar do fluxo espectral e uma forma automática de obtenção do valor ótimo para o ganho do limiar.

No Capítulo 4, o algoritmo abordado trata da obtenção da melodia de uma peça musical, considerando um sinal monofônico. Essa tarefa foi realizada por meio da função de saliência independente do timbre. Uma pequena modificação do algoritmo foi feita, ao aplicarmos o SSE nas raias espectrais para estimar os candidatos a f_0 de forma mais acurada.

Depois, foi proposta, no Capítulo 5, a separação de componentes tonais e transitórias como uma maneira de melhorar o desempenho dos algoritmos descritos anteriormente. O método utilizado foi a filtragem por mediana.

No Capítulo 6, foram apresentados os resultados de alguns experimentos, usando o método de STFT, apresentado no Capítulo 2. Para verificar a validade da proposta,

foram realizados experimentos, considerando o comprimento do filtro de mediana $L \in \{10, 20, 30\}$. Como resultado, foi observado um aumento nos valores médios das medidas de desempenho. A Medida-F, empregada na detecção de *onsets* aumentou em $\{9,83\%;\ 19,35\%;\ 20,79\%\}$, respectivamente. Já o algoritmo de rastreamento de f_0 teve a sua Precisão média aumentada em $\{7,04\%;\ 11,43\%;\ 13,18\%\}$. Visto que uma variação de 10% foi estabelecida como valor mínimo para que o resultado seja considerado aceitável, concluiu-se que, para ambos os algoritmos, o filtro de $L = 10$ amostras não é recomendado. Além disso, levando em consideração o aumento no custo computacional, temos que o filtro de comprimento igual a 30 amostras não é tão efetivo. Assim sendo, o filtro de $L = 20$ amostras é o mais recomendado para os dois algoritmos.

7.2 Trabalhos futuros

As próximas etapas para a continuação desse trabalho incluem, primeiramente, a verificação da influência da separação de componentes, considerando os outros métodos de obtenção da representação tempo-frequencial descritos no Capítulo 2. Além disso, existem modificações a serem feitas nos algoritmos descritos nos Capítulos 3 e 4.

Para a tarefa de detecção de *onsets*, é possível adicionar outras etapas de pré-processamento e conferir se esses procedimentos produzem melhores resultados, quando utilizados em conjunto com o filtro de mediana. Por exemplo, em [12], é sugerido o uso de filtros Mel, que fazem um mapeamento do espectro do sinal em uma escala não linear.

Em relação à tarefa de rastreamento de f_0, existem dois pontos principais a serem melhorados. Primeiramente, temos que o algoritmo utilizado associa a todos os quadros uma f_0, não considerando períodos de silêncio. Isso leva a uma quantidade considerável de erros. Uma possível forma de contornar esse problema é estabelecer um valor mínimo de saliência para que os picos sejam considerados. Outro ponto a ser melhorado é o fato de que esse algoritmo trata de sinais monofônicos, que não é o caso da maioria das peças musicais. É necessário, portanto, generalizar o método, de forma a considerar a ocorrência simultânea de múltiplas frequências fundamentais.

Por fim, cabe o estudo de outros métodos para realizar as tarefas descritas nesse trabalho. Ao analisarmos publicações mais recentes na área de Extração de Informação Musical, é possível notar que o estado-da-arte de diversas tarefas tem se encaminhado

para soluções baseadas em aprendizado de máquina [25, 26].

Referências Bibliográficas

[1] MÜLLER, Meinard. *Fundamentals of Music Processing*. Berlim, Alemanha, Springer Verlag, 2015.

[2] OPPENHEIM, Alan V.; SCHAFER, Ronald W.; BUCK, John R. *Discrete-Time Signal Processing*. Nova Jérsei, EUA, Prentice-Hall, 1998.

[3] KLAPURI, Anssi et al. *Signal Processing Methods for Music Transcription*. Nova Iorque, EUA, Springer, 2006.

[4] BROWN, Judith C. "Calculation of a constant Q spectral transform". *Journal of the Acoustical Society of America*, vol. 80, n. 1, pp. 425–434, janeiro, 1991.

[5] WERUAGA, Luis; KÉPESI, Márian. "The fan-chirp transform for non-stationary harmonic signals". *Signal Processing*, vol. 87, n. 6, pp. 1504–1522, junho, 2007.

[6] CANCELA, Pablo; LÓPEZ, Ernesto; ROCAMORA, Martín. "Fan-chirp transform for music representation". *Proceedings of the 11th International Conference on Digital Audio E ects (DAFx-10)*, pp. 1–8, Graz, Áustria, setembro, 2010.

[7] LAROCHE, Jean. "Efficient tempo and beat tracking in audio recordings". *Journal of the Audio Engineering Society*, vol. 51, n. 4, pp. 226–233, abril, 2003.

[8] BELLO, Juan Pablo et al. "A tutorial on onset detection in musical signals". *IEEE Transactions on Speech and Audio Processing*, vol. 13, n. 5, pp. 1035–1047, 2005.

[9] DIXON, Simon. "Onset detection revisited". *Proceedings of the 9th International Conference on Digital Audio E ects (DAFx-06)*, pp. 133-137, Montreal, Canadá, setembro, 2006.

[10] LEVEAU, Pierre; DAUDET, Laurent; RICHARD, Gaël. "Methodology and tools for the evaluation of automatic onset detection algorithms in music". *Proceedings of the*

International Conference on Music Information Retrieval (ISMIR'04), pp. 72–75, Barcelona, Espanha, outubro, 2004.

[11] LAURENTI, Nicola; DE POLI, Giovanni; MONTAGNER, Daniele. "A nonlinear method for stochastic spectrum estimation in the modeling of musical sounds". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 2, n. 15, pp. 531–541, fevereiro, 2007.

[12] NUNES, Leonardo O.; BISCAINHO, Luiz Wagner P. "Tempo estimation: evaluation of different spectral flux computation methods". *Anais do 10º Congresso / 16ª Convencao Nacional da AES Brasil (AES-Brasil 2012)*, maio, 2012.

[13] DEGANI, Alessio; PEETERS, Geoffroy. "A pitch salience function derived from harmonic frequency deviations for polyphonic music analisys". *Proceedings of the 17th Conference on Digital Audio E ects (DAFx-14)*, Erlangen, Alemanha, setembro, 2014.

[14] SALAMON, Justin; GÓMEZ, Emilia; BONADA, Jordi. "Sinusoid extraction and salience function design for predominant melody estimation". *Proceedings of the 14th Conference on Digital Audio E ects (DAFx-11)*, Paris, França, setembro, 2011.

[15] AMATRIAIN, Xavier et al. "Spectral Processing". ZÖLZER, Udo et al. *DAFX-Digital Audio E ects*, cap. 10, pp. 373-438, John Wiley & Sons, 2002.

[16] APOLINÁRIO, Isabela F. *Contribuicoes a Metodos para Representacao Tempo-Frequencial de Sinais de Musica*, dissertação de Mestrado, PEE/COPPE/UFRJ, setembro, 2015.

[17] ESQUEF, Paulo; BISCAINHO, Luiz Wagner P. "Spectral-based analysis and synthesis of audio signals". PEREZ-MEANA, Hector et al. *Advances in Audio and Speech Signal Processing: Technologies and Applications*, cap. 3, pp. 56-92, Hershey, EUA, fevereiro, 2007.

[18] SALAMON, Justin et al. "An analysis/synthesis framework for automatic f0 annotation of multitrack datasets". *18th International Society for Music Information Retrieval Conference (ISMIR'17)*, Suzhou, China, outubro, 2017.

[19] BITTNER, Rachel M. et al. “MedleyDB: A multitrack dataset for annotation-intensive MIR research”. *15th International Society for Music Information Retrieval Conference (ISMIR'14)*, pp. 155–160, Taipei, Taiwan, outubro, 2014.

[20] YOSHII, Kazuyoshi; GOTO, Masataka; OKUNO, Hiroshi G. “Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression”. *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, n. 1, pp. 333-345, janeiro, 2007.

[21] GILLET, Olivier; RICHARD, Gaë. “Transcription and separation of drum signals from polyphonic music”. *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, n. 3, pp. 529-540, março, 2007.

[22] FITZGERALD, Derry. “Harmonic/percussive separation using median filtering”. *Proceedings of the 13th Conference on Digital Audio E ects (DAFx-10)*, Graz, Áustria, setembro, 2010.

[23] ONO, Nobutaka et. al. “Separation of monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram”. *Proceedings of the EUSIPCO 2008 European Signal Processing Conference*, Lausanne, Suíça, agosto, 2008.

[24] JAIN, Ramesh; KATSURI Rangachar; SCHUNCK, Brian. *Machine Vision*. EUA, McGraw-Hill, 1995.

[25] KIM, Jong Wook; BELLO, Juan Pablo. “Adversarial learning for improved onsets and frames music transcription”. *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'19)*, pp. 670-677, Delft, Holanda, novembro, 2019.

[26] CUESTA, Helena; MCFEE, Brian; GÓMEZ, Emilia. “Multiple f0 estimation in vocal ensembles using convolutional neural networks”. *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR'20)*, pp. 302-309, Montreal, Canadá, outubro, 2020.